WHAT IS CLAIMED IS:

1. A method comprising the steps of:

   (a) providing a database of rules;

   (b) applying an algorithm to the database to identify Almost-Exact Rules and Other Rules;

   (c) partitioning the database so that the Almost-Exact Rules are grouped into one or more groups;

   (d) partitioning the database so that the Other Rules are grouped in at least one separate group.

2. The method of Claim 1 further including the step of using FM search algorithm to test packets with the Almost-Exact rules in the one or more groups.

3. The method of claim 1 further including the step of using an SMT algorithm to test packets with the Other rules in the separate group.

4. The method of claim 1 further including the step of using a Content-Addressable Memory (CAM) to test packets with the other rules in the separate group.

5. The method of claim 1 wherein the database of rules is being partitioned as a function of

fields within each rules.

6. A Network Processor comprising:

a first database storing filter rules or other classification rules that are exact in all fields except one;

a second database storing other filter rules or other classification rules;

a first search function receiving an IP packet and testing a portion of said packet against the first database;

a second search function receiving an IP packet and testing a portion of said packet against the second database; and

an Arbitrator function responsive to signals from the first search function or the second search function to output an action signal if a match is found.

7. The Network Processor of Claim 6 wherein the first search function includes a Full Match (FM) algorithm.

8. The Network Processor of Claim 6 wherein the second search function includes a Software Managed Tree (SMT) algorithm.

9. The Network Processor of Claim 6 further including a third search function receiving an IP packet and test a portion of the packet against the second database.

10. The Network Processor of Claim 9 wherein the third search function includes Content-Addressable Memory.

11. The Network Processor of Claim 6 further including a control processor operatively connected to the Network Processor wherein said control processor is programmed to generate the first database and the second database.

12. The Network Processor of Claim 6 wherein the first database and the second database are partitioned from a common database.

13. A program product comprising:

media on which computer instructions are recorded, said instructions including a first code module that parses database of rules and partitions said database into n sets, wherein n represents number of fields in each rule of said database;

a second code module that interrogates the n sets and deletes from each set rules not meeting a first predetermined criteria;

a third code module that interrogates remaining rules in each set S_i, i = 1, 2, ..., n, to determine said remaining rules are what fraction f_i of the rules in the database;

a fourth code module that interrogates f_i for each set Si; and

grouping rules associated with f_i, if said f_i meets a second predetermined criteria, into one or more groups and other rules into at least one separate group.

14. The program product of Claim 13 wherein the one or more groups include Almost-Exact rules defined relative to a chosen field i.

15. The program product of Claim 14 wherein the separate group includes all other rules.

16. The program product of Claim 14 further including a Full Match (FM) algorithm that tests a key against rules in the one or more groups.

17. The program product of Claim 14 wherein a Software Managed Tree (SMT) algorithm tests the key against rules in said at least one separate group.

18. The program product of Claim 14 wherein a Content-Addressable Memory tests the key against rules in said at least one separate group.

19. A method comprising the acts of:

providing a database of rules;

partitioning, with an algorithm, said database of rules into n sets, where n represents number of fields in each rule;

reducing the number of rules within each set based upon characteristics of fields within each rule;

for remaining rules in each set, S_i, with i-1, 2, ..., n, calculate a fraction fi, wherein

$$fi = \frac{\text{Number of Rules in set } S_i}{\text{Total Number of Rules In Database}} \; ;$$

setting a predetermined threshold T;

if fi meets or exceeds the predetermined threshold T, then partitioning rules into at least one group S_i and all other rules into at least one separate group.

20. The method of Claim 19 further including the act of using a Full Match (FM) algorithm to test a key against rules in the at least one group.

21. The method of Claim 19 further including the act of using a Software Managed Tree (SMT) algorithm to test a key against rules in the at least separate group.

22. The method of Claim 19 further including the act of using a Content-Addressable Memory algorithm to test a key against rules in the at least separate group.

23. The method of Claim 19 wherein the act of partitioning includes testing of the i^{th} field of each rule and only allowing to remain the rules with a wild-card specification in field i

within the set S_i of almost-exact rules.

24. The method of Claim 19 or 23 wherein the act of reducing further includes the acts of determining rules with non-exact fields; and

deleting said rules with non-exact fields from each set.

25. The method of Claim 21 further including the acts of determining rules in each set that intersect with any other rule in the database of rules that has higher priority; and

deleting intersecting such rules from each set.